© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 07 Issue: 04 | July – September 2016
Paper is available at www.jrps.in | Email : info@jrps.in

Study of error detection and correction codes

[1]Seema Rani, lohanpoonam@gmail.com
[2]Amita , Assistant professor , Govt. College, Julana

Introduction : Where there are inputs and a corresponding outputs error is ominous. Similarly in case of digital systems in various cases be it a digital computer or a digital communication set up, error occurrence is a common phenomenon. And for that the first step is to detect the error and after that errors are corrected. The most common cause for errors are that the noise creep into the bit stream during the course of transmission from transmitter to the receiver. And if these errors are not detected and corrected the result could be disastrous as the digital systems are very much sensitive to errors and will malfunction due to the slightest of errors in transmitted codes. There are various methods of error detection and correction such as addition of extra bits which are also called check bits, sometimes they are also called redundant bits as they don't have any information in them. In this article we will discuss about the various codes which are used for error detection and correction code in digital system.

An error-correcting code uses multiple parity check bits that are stored with the data word in memory. Each check bit is a parity bit for a group of bits in the data word. When the word is read back from memory, the parity of each group, including the check bit, is evaluated. If the parity is correct for all groups, it signifies that no detectable error has occurred. If one or more of the newly generated parity values is incorrect, a unique pattern called a syndrome results that may be able to identify which bit is in error. A single error occurs when a bit changes in value from 1 to 0 or from 0 to 1 while stored or if it erroneously changes during a write or read operation. If the specific bit in error is identified, then the error can be corrected by complementing the erroneous bit.

Parity Code

Parity bit is added to the transmitted strings of bits during transmission from transmitters to detect any error in the data when they are received at the receiver end. Basically a parity code is nothing but an extra bit added to the string of data. Now there are two types of parity these are even parity and odd parity. Now we get an even parity when the total numbers of 1's in the string of the data is even after adding that extra bit. Similarly we get an odd parity when after adding that extra bit into the data string the total number of 1's in the data is odd. We can understand it with an example, suppose we have an eight bit ASCII code – 01000001. Now if the added bit is 0 then the number will become 001000001. Here the total number of 1s in the number is even so we get an even parity. Again if we add 1 to the number the number will become 101000001. Here the number of 1s is 3 which is odd so, we have got an odd parity. Normally even parity is used and it has almost become a convention. Now parity checks are capable of detecting a single bit error but it fails if there are two changes in the data and it is the biggest drawback of this system. That's why there are several other codes to detect and correct more than one bit errors.

Repetition Code

In repetition code a single bit data is considered as a bit string of predetermined value and sent to the receiver, this is capable of detecting more than one data bit error. This can be illustrated with an example suppose the original number is 101. Now during transmission all the numbers are repeated say 3 times, so the final transmitted number is 111000111. So when the number is received 1 bit error and two bit errors can be easily identified like it will be 011, 110 or 101. So it is a better way to detect and correct data but it gets highly inefficient as the number of repeated bits increase.

Cyclic Redundancy Check Code

Cyclic redundancy check (CRC) codes provide a reasonably high level of protection at low redundancy Level. The cycle code for a given data word is generated as follows. At first we have to add certain number zeroes (the numbers are determined by the desired number of bit checks. This new data bit sequence is then divided by a special binary word whose length equals n + 1, n being the number of check bits to be added. The remainder obtained as a result of modulo-2 division is then added to the dividend bit sequence to get the cyclic code. The code word generated after the operation is completely divisible by the divisor which was used in the generation of the code. Thus, when we divide the received code with the same divisor, an error-free reception should lead to an all '0' remainder. A nonzero remainder is indicative of the presence of errors. The probability of error detection depends upon the number of check bits, n, used to construct the cyclic code. It is 100 % for single-bit and two-bit errors. It is also 100 % when an odd number of bits are in error and the error bursts have a length less than n + 1. The probability of detection reduces to $1 - (1/2)^{n-1}$ for an error burst length equal to n + 1, and to $1 - (1/2)^{n}$ for an error burst length greater than n + 1.

Hamming Codes : The most common types of error-correcting codes used in RAM are based on the codes devised by R. W. Hamming. In the Hamming code, k parity bits are added to an n-bit data word, forming a new word of n k bits. The bit positions are numbered in sequence from 1 to n k. Those positions numbered with powers of two are reserved for the parity bits. The remaining bits are the data bits. The code can be used with words of any length. Before giving the general characteristics of the Hamming code, we will illustrate its operation with a data word of eight bits. Consider, for example, the 8-bit data word 11000100. We include four parity bits with this word and arrange the 12 bits as follows: The 4 parity bits P1 through P8 are in positions 1, 2, 4, and 8, respectively. The 8 bits of the data word are in the remaining positions. Each parity bit is calculated as follows: P1 XOR of bits (3, 5, 7, 9, 11) P2 XOR of bits (3, 6, 7, 10, 11) P4 XOR of bits (5, 6, 7, 12) P8 XOR of bits (9, 10, 11, 12) Recall that the exclusive-OR operation performs the odd function. It is equal to 1 for an odd number of 1's among the variables and to 0 for an even number of 1's. Thus, each parity bit is set so that the total number of 1's in the checked positions, including the parity bit, is always even. The 8-bit data word is written into the memory together with the 4 parity bits as a 12-bit composite word.

A modified Hamming code to generate and check parity bits for a singleerror-correction, double-error-detection scheme is most often used in real systems. The modified code uses a different parity check bit scheme that balances the number of inputs to the logic for each check bit and thus the number of inputs to each circuit that does the checking. The balancing minimizes the delay through the error correction and detection circuits. These circuits can be used in a RAM subsystem to add check bits during write operations and to correct single errors and detect double errors during read operations.

References :

1. https://en.wikipedia.org/wiki/Error_detection_and_correction
2. http://www.tutorialspoint.com/computer_logical_organization/error_codes.htm
3. http://www.lee.eng.uerj.br/~gil/redesII/hamming.pdf
4. http://logos.cs.uic.edu/366/notes/ErrorCorrectionAndDetectionSupplement.pdf
5. http://www.electrical4u.com/error-detection-and-correction-codes/